USAC
TRICENTENARIA
Universidad de San Carlos de Guatemala

Manual técnico

Juan José Gerardi Hernandez
carné: 201900532

Descripción de la práctica: se solicita un programa el cual analiza la entrada en lenguaje dataforge el cual es capaz de realizar operaciones aritméticas como también generar gráficas.

Librerías utilizadas:

- ➔ CUP: Constructor Basado de Parsers Útiles (CUP para abreviar). CUP es un sistema para generar analizadores LALR a partir de especificaciones simples. Sirve el mismo papel que el programa ampliamente utilizado YACC y de hecho ofrece la mayoría de las características de YACC. Sin embargo, CUP está escrito en Java, utiliza especificaciones que incluyen incrustado Código Java, y produce analizadores que se implementan en Java.

- ➔ JFLEX: JFlex es un generador de analizador léxico para Java escrito en Java. es también una reescritura de la herramienta JLex (Berk 1996) que fue desarrollado por Elliot Berk en la Universidad de Princeton. Como Vern Paxson afirma para su flexión de herramientas C / C + + (Paxson 1995): ellos hacen sin embargo, no comparten ningún código.
  Un generador de analizador léxico toma como entrada una especificación con un conjunto de expresiones regulares y acciones correspondientes. Genera un programa (a *lexer*) que lee la entrada, coincide con la entrada en contra las expresiones regulares en el archivo de especificaciones, y ejecuta la correspondiente acción si una expresión regular coincide. Los Lexers generalmente son los primeros pasos de front-end en compiladores, palabras clave coincidentes, comentarios, operadores, etc, etc., y generan un flujo de token de entrada para analizadores. También pueden ser utilizado para muchos otros propósitos.

funciones de operaciones:

```java
public static double Media(ArrayList<Double> lista){
    double sumatoria = 0;
    double promedio;
    for (int i=0;i<lista.size();i++) {
        sumatoria = sumatoria+lista.get(i);
    };
    promedio = sumatoria/lista.size();
    promedio = Math.floor(promedio*100)/100;
    return promedio;
}
```

```java
public static double Moda(ArrayList<Double> lista){
    HashMap<Double,Integer> frec = new HashMap<>();
    for(Double valor:lista){
        if (frec.containsKey(valor)){
            frec.put(valor,frec.get(valor)+1);
        }else{
            frec.put(valor,1);
        }
    }
    Double moda = null;
    int frecMax = 0;
    for (Map.Entry<Double,Integer> entry : frec.entrySet()){
        if(entry.getValue() > frecMax){
            moda = entry.getKey();
            frecMax = entry.getValue();
        }
    }
    return moda;
}
```

```java
    public static double Mediana(ArrayList<Double> lista){
        Collections.sort(lista);
        int n = lista.size();
        if (n%2!=0){
            return lista.get(n/2);
        } else {
            double v1 = lista.get((n/2)-1);
            double v2 = lista.get((n/2));
            return (v2+v1)/2.0;
        }
    }
```

```java
public static double Varianza(ArrayList<Double> lista){
    double suma = 0;
    int n = lista.size();
    for(int i=0;i<lista.size();i++){
        suma+=lista.get(i);
    }
    double media = suma/n;
    double sumacua =0.0;
    for (int j=0;j<lista.size();j++){
        sumacua += Math.pow(lista.get(j)-media,2);
    }
    double varianza = sumacua/n;
    return varianza;
}
```

```java
public static Double Max(ArrayList<Double> lista){
    Collections.sort(lista);
    int n = lista.size();
    return lista.get(n-1);
}
1 usage  Juan Gerardi
public static Double Min(ArrayList<Double> lista){
    Collections.sort(lista);
    int n = lista.size();
    return lista.get(0);
}
```